**Animatlab:Matlab Virtual Serial Communication**

Credit to Wade Hilts for figuring this all out in the first place.

Also in folder:

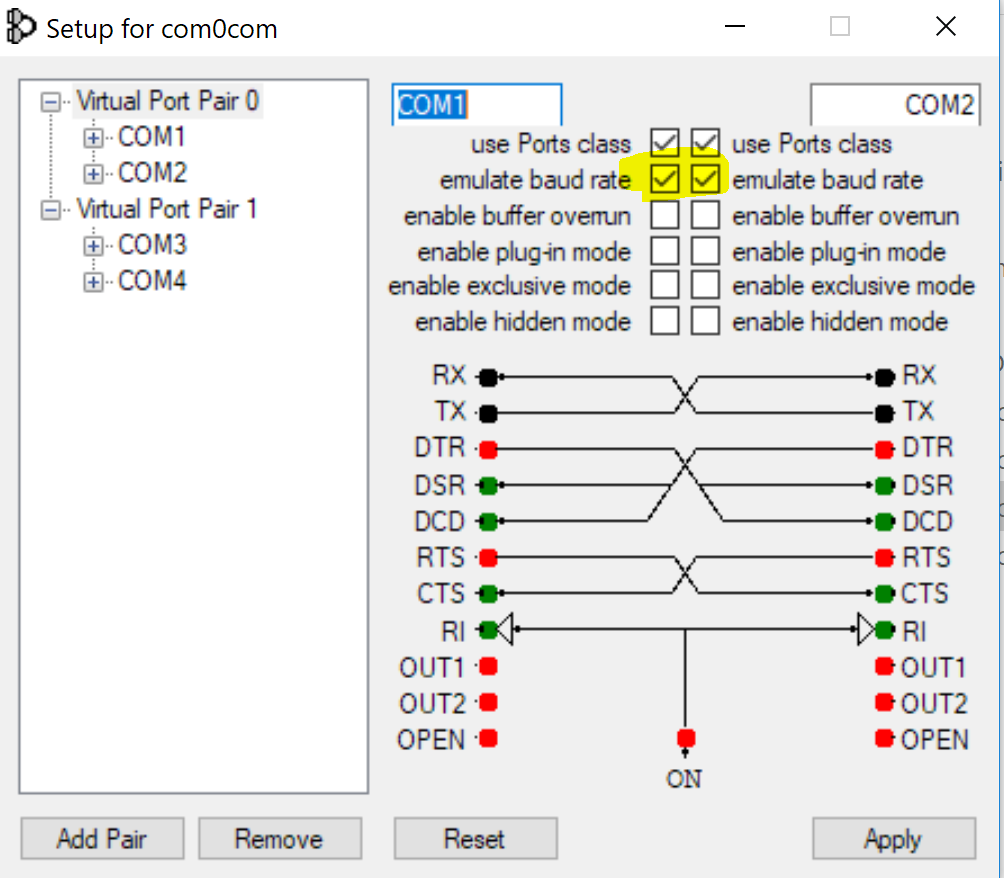
* See document [AnimatSerial\_ICD.docx](http://animatlab.com/Portals/0/Media/AnimatSerial_ICD.docx) published by Animatlab about setting up serial
* Note that there are several more semi-related resources on the Animatlab website, but they are not well organized

Download necessary software:

* Virtual Port Creator such as [com0com](https://sourceforge.net/projects/com0com/)
  + Critical feature is “Baudrate Emulator” - for instance “Free Virtual Serial Ports” does not have this, and doesn’t seem to work in the AM to Matlab direction

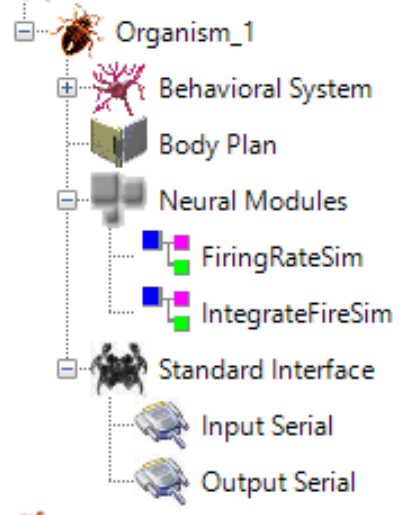
Use Virtual Port Creator to set up two serial bridges

* Identify which port you want to communicate from Matlab to Animatlab and vice versa
* Note that for each bridge, a certain COM number is associated with Matlab while a different COM number is associated with Animatlab
* Select “enable baud rate”

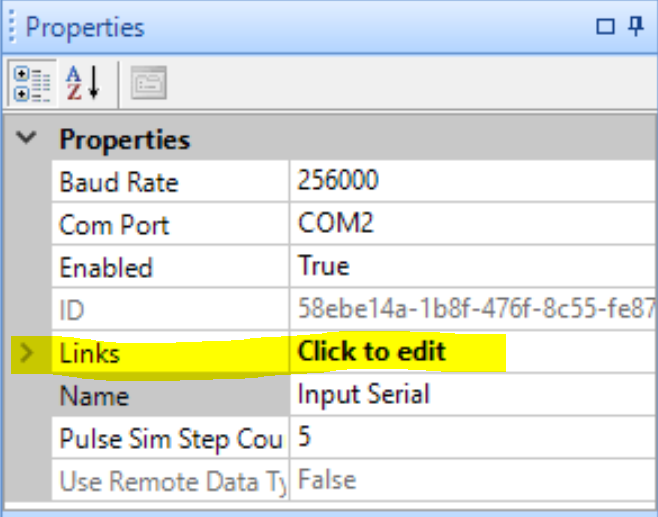


Prepwork in Animatlab

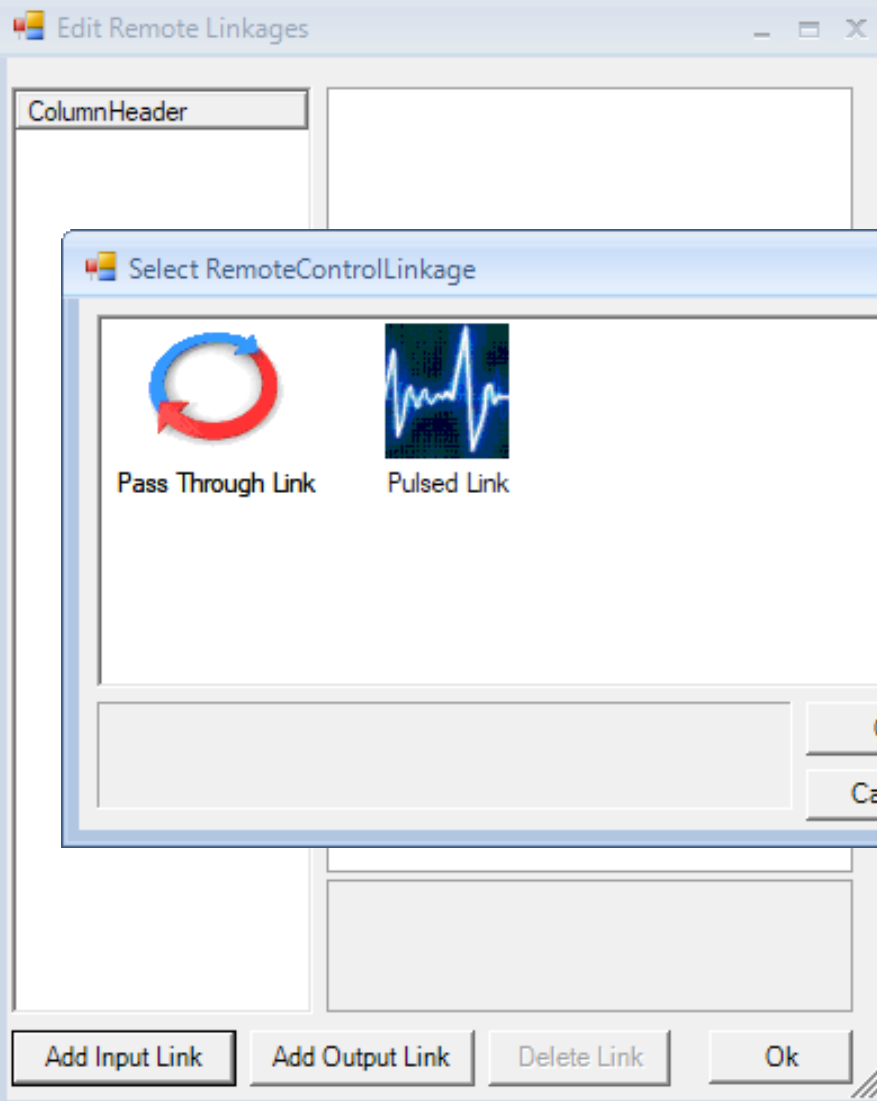
* Create AM Neural Network
* Set up [Robot Interface](http://animatlab.com/Help/Documentation/Robotics/Robot-Interfaces)
  + Right click on Organism in the treeview and select Add robot interface
  + Choose Standard Interface
    - Right click on Standard Interface in treeview and select Add IO Control
      * Select Animat Serial
      * Rename to Input or Output serial
    - Repeat to have one Input and one Output channel



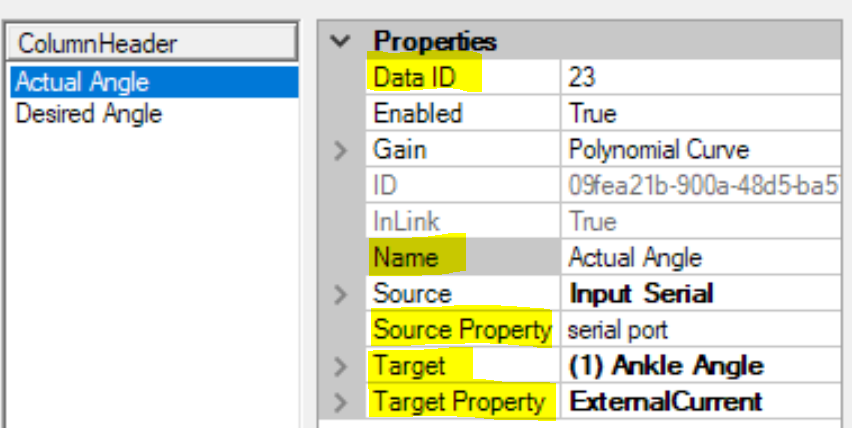
* Update Input and Output Serial Properties
  + Set Baudrate (same rate designated in Matlab, AM, and Virtual Port)
  + Set correct COM ports to each channel as determined previously
  + Select “Links” to identify what specific neuron properties are to be transferred over serial



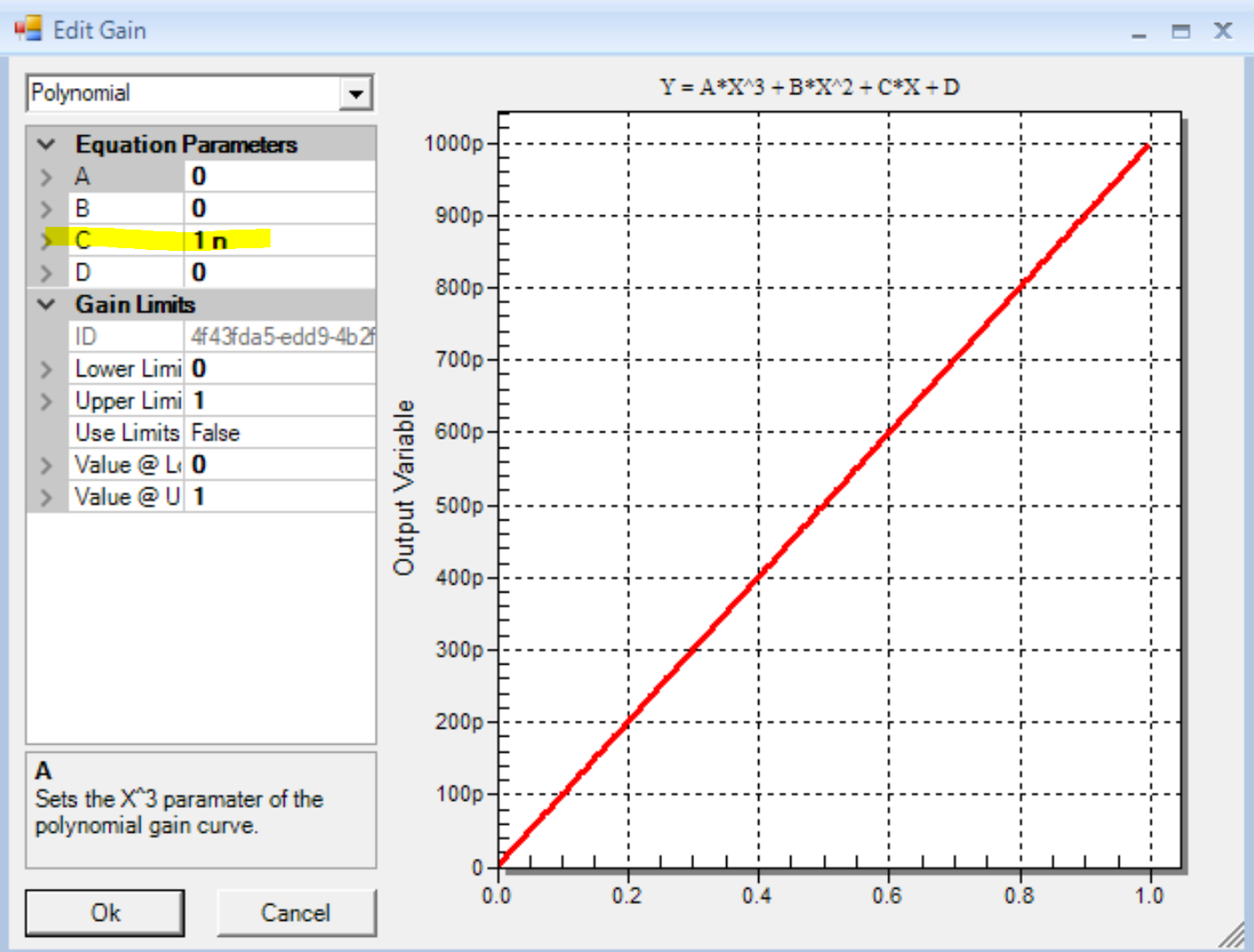
* + In Links menu, select Add Link and select Pass Through Link for continuous applications



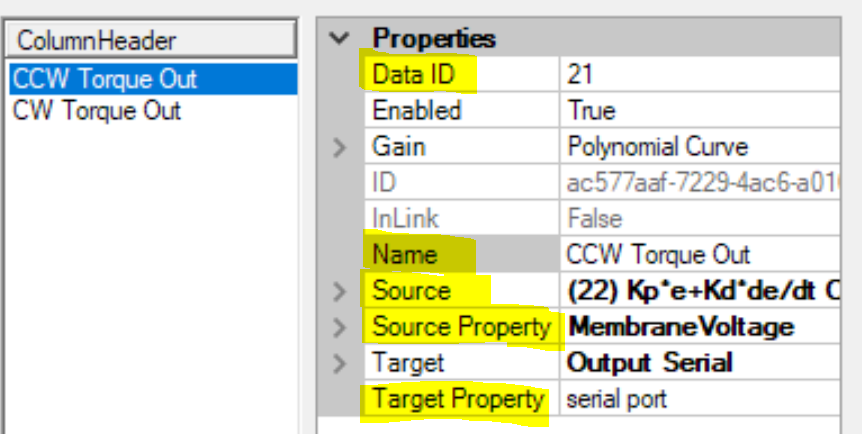
* + Enter properties of data for transfer over serial
    - Create Data ID that will be referenced in Matlab
    - Name the Data
    - For Input
      * Name Source Property (doesn’t matter what)
      * Identify Target Neuron (where does the signal from Matlab go?)
      * Identify Target Property (what property to update from Matlab)



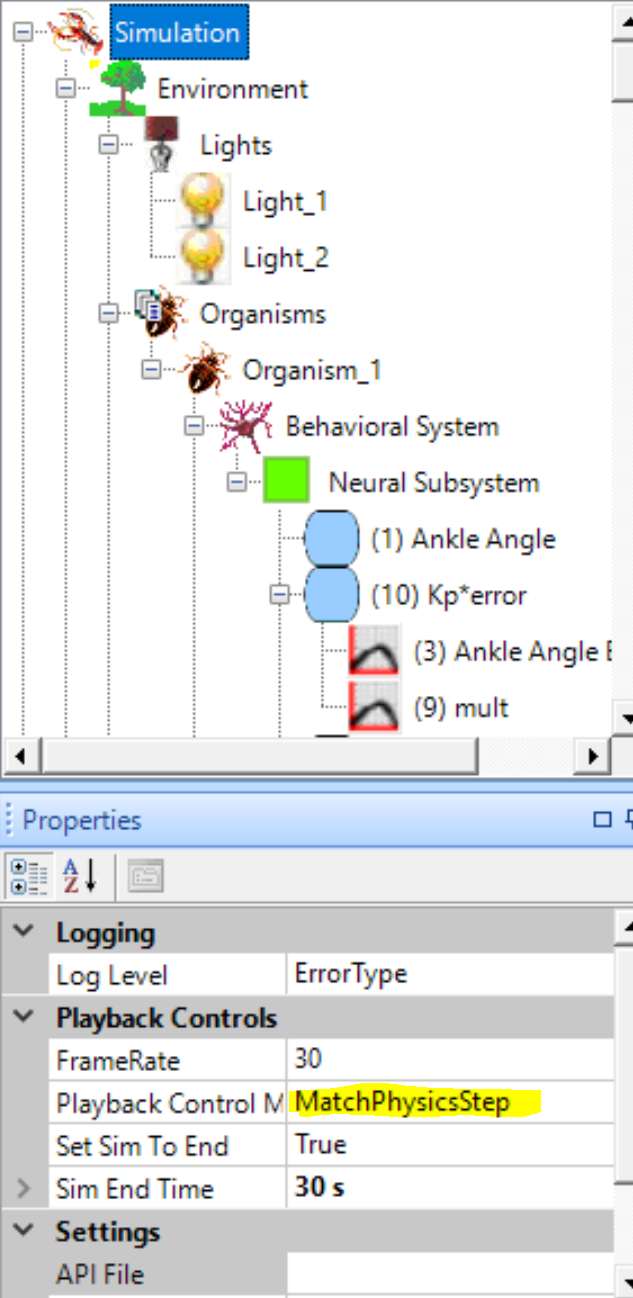
* + - * For scaling of data inputs to Animatlab, enter “Gain - Polynomial Curve” menu.
        + For linear gain, set C parameter to desired scaling
        + For instance, if input of 1 from Matlab should be an input of 1nA to Animatlab, set C to 1 n as shown



* + - For Output
      * Name Target Property (doesn’t matter what)
      * Identify Source Neuron (what data to transfer to Matlab)
      * Identify Source Property (what property to transfer to Matlab)
      * For scaling of data outputs from Animatlab, enter “Gain - Polynomial Curve” menu and set the same as with inputs



* Update Simulation Properties
  + Click on Simulation in tree
  + Set Playback Control Mode to Match Physics step
    - If set to FastestPossible, simulation will run faster than serial comm, so you will miss inputs and outputs, but it will look like things are happening!
  + “Set Sim to End” = True if desired, in which case set Sim Time
    - If not set, simulation will run until the user stops it



Prepwork in Matlab **(see sample code in folder)**

* Open two serial ports
* Set up serial reads by reading, identifying and processing one byte at a time as per the packet structure referenced in AnimatSerial\_ICD doc
* Set up serial writes using packet structure referenced in AnimatSerial\_ICD doc

How to Make it Go

* Click “Run” in Matlab first (since Matlab opens the serial ports, it has to initiate first)
* Then click “Play” in Animatlab
* A “Timeout” error in Matlab means that Animatlab has stopped sending data. This will happen if your code has no end condition and the simulation time ends, or if the simulation hangs for some other reason.
  + Animatlab only sends data if the Output value has changed

Troubleshooting

* Problem: “Timeout” error happens too fast, i.e. after only 1 or two packets have transferred
  + Solution: sometimes it works to add a tiny value for Matlab to send to Animatlab on one of the input channels - i.e., excite the Animatlab side of things enough to make sure data values on the outputs change within the first data cycle.

Things to Know

* If the simulation is running between Matlab and Animatlab (or some other hardware and Animatlab), the simulation time must agree on both sides. If not, Animatlab will misinterpret the frequency scale, and will not filter the signal at the proper frequencies, resulting in incorrect frequency response and plots.
* If simulation timing is mismatched between the two hardwares, the Robot Timestep in particular, more serial errors will occur.
* When adding a new serial connection, bear in mind that Animatlab chooses its own order to send data, and it is unclear what that order is. The order is not numerical, i.e., it does not necessarily send channel 21 then 22 then 23, etc. It also does not follow the order listed in the Input or Output dialog box. When you add a new serial connection, save, close, and reopen Animatlab so it chooses its sending order before writing your code. Then read in the raw data to see the order and adjust your code accordingly.
  + Once AM chooses its order, it seems to stick to it, but it can change any time you add a new serial connection.
  + Ideally, code deals with the serial connections by ID only and does not rely on a certain order.